



(12) **DEMANDE DE BREVET EUROPEEN**

(43) Date de publication:
22.11.2000 Bulletin 2000/47

(51) Int Cl.7: **G06F 17/30**

(21) Numéro de dépôt: **00401375.1**

(22) Date de dépôt: **19.05.2000**

(84) Etats contractants désignés:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE
 Etats d'extension désignés:
AL LT LV MK RO SI

(71) Demandeur: **BULL S.A.**
78434 Louveciennes Cedex (FR)

(72) Inventeur: **Caubit, Pierre**
49000 Angers (FR)

(30) Priorité: **19.05.1999 FR 9906353**

(54) **Système et procédé de gestion d'attributs dans un environnement orienté objet**

(57) La présente invention concerne un procédé de gestion d'une base de données (4) ou annuaire ou équivalent comportant des objets (5) caractérisés par des attributs standards (6), la base étant apte à être interrogée par une application, un utilisateur ou autre, appelé entité, consistant à caractériser les objets (5) par des

attributs spécifiques (20) dits attributs opaques (20), présentant la même structure externe que les attributs standards (6) et dont la structure interne permet de définir et administrer des composants (21) de manière indépendante à la base (4).

La présente invention concerne également le système mettant en oeuvre ledit procédé.

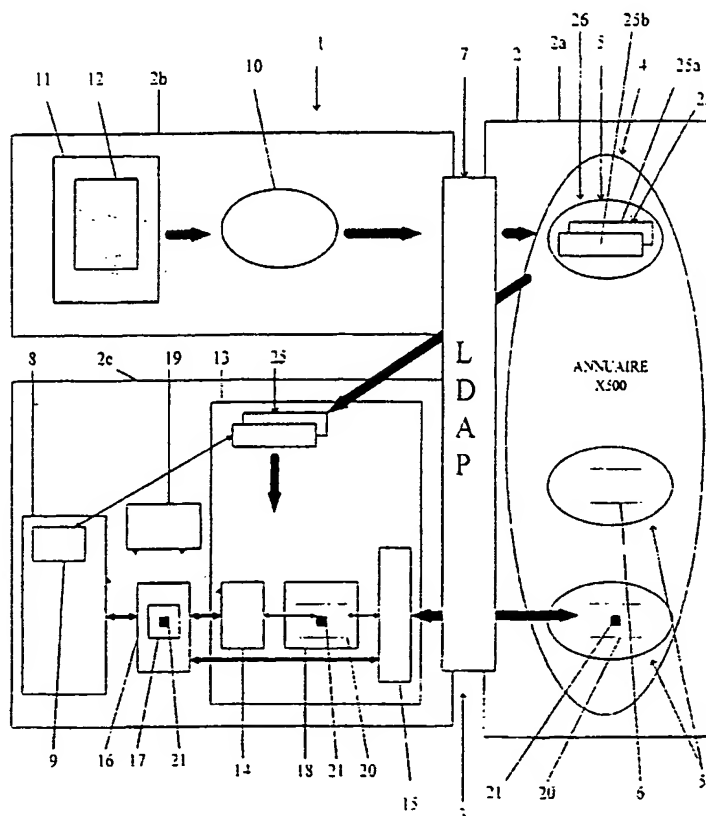


FIG.1

Description

[0001] La présente invention concerne le domaine des bases de données dans un environnement d'ordinateur orienté objet.

L'art antérieur

[0002] Une base de données, un annuaire ou équivalent organise de manière cohérente des objets partagés par des utilisateurs ou des applications.

[0003] Un objet est une abstraction d'une donnée du monde réel. L'objet est caractérisé par une identité, un état et un comportement. L'identité d'un objet est représentée par un identifiant unique et invariant qui permet de référencer l'objet indépendamment des autres objets. Les valeurs de l'état d'un objet sont données par des attributs. Le comportement d'un objet est défini par un ensemble d'opérations applicables à l'objet. Les objets similaires sont groupés en classe. Une classe décrit la structure et le comportement commun d'objets. Les objets et les attributs sont définis dans un schéma de la base.

[0004] Dans les bases de données actuelles, tous les attributs sont déclarés dans le schéma de la base.

[0005] Or, la complexité et l'évolution rapide des informations contenues dans une base de données soulèvent de nombreux problèmes :

[0006] La description du schéma de la base est fréquemment modifiée ce qui nécessite une gestion permanente et coûteuse de la base et entraîne une gêne pour les utilisateurs et administrateurs.

[0007] Des objets de versions différentes sont susceptibles d'être gérés par une même base. Le problème de gestion de versions se complique avec la réplication des bases dans un environnement distribué et plus particulièrement lorsque les différentes versions se situent sur des machines différentes.

[0008] L'utilisateur a souvent besoin d'accéder rapidement à une suite d'attributs d'un objet donné. La base est sollicitée pour la recherche de chacun des attributs. Il en résulte un encombrement dans la consultation de la base, une lenteur dans l'obtention d'une réponse à une interrogation.

[0009] Il serait avantageux que des composants de types différents (chaîne de caractères, entier, booléen...) soient stockés dans un même attribut, ce qui n'est pas réalisable avec les systèmes actuels connus.

[0010] L'origine, l'intégrité ou la confidentialité des informations stockées dans la base ne sont garanties qu'après application de techniques diverses sur les attributs, techniques rapportées, lourdes et coûteuses.

[0011] Un but de la présente invention est de permettre de modifier facilement et rapidement les informations associées à une classe d'objets.

[0012] Un autre but de la présente invention est d'accéder rapidement à une suite d'attributs.

[0013] Un autre but est d'intégrer des composants de

différents types dans un même attribut.

[0014] Un autre but de la présente invention est d'administrer la cohabitation d'objets de versions différentes dans une même base ou des bases répliquées.

5 [0015] Un autre but est de garantir l'origine, l'intégrité et la confidentialité de l'information sans recours à des techniques rapportées.

Résumé de l'invention

10 [0016] Dans ce contexte, la présente invention propose un procédé de gestion d'une base de données ou annuaire ou équivalent comportant des objets caractérisés par des attributs standards, la base étant apte à être interrogée par une application, un utilisateur ou autre appelé entité, caractérisé en ce qu'il consiste à caractériser les objets par des attributs spécifiques dits attributs opaques, présentant la même structure externe que les attributs standards et dont la structure interne permet de définir et administrer des composants de manière indépendante à la base.

15 [0017] La présente invention concerne le système informatique de mise en oeuvre du procédé selon l'invention.

20 [0018] La présente invention se rapporte également à un système informatique comprenant au moins une machine munie d'une base de données ou annuaire ou équivalent susceptible d'être interrogée par une application, un utilisateur ou autre, appelé entité, caractérisé en ce qu'il comprend une interface de programmation au niveau applicatif API entre la base et l'entité comprenant des fonctions d'accès à la base et des fonctions de gestion d'attributs standards et/ou d'attributs opaques.

Présentation des figures

25 [0019] D'autres caractéristiques et avantages de l'invention apparaîtront à la lumière de la description qui suit, donnée à titre d'exemple illustratif et non limitatif de la présente invention, en référence aux dessins annexés dans lesquels:

- la figure 1 est une vue schématique du système selon la présente invention ;
- la figure 2 représente de manière schématique la structure d'un attribut opaque du système selon la figure 1 ;
- la figure 3 représente de manière schématique des étapes du procédé selon la présente invention.

Description d'une forme de réalisation de l'invention

30 [0020] Comme représenté sur la figure 1, la présente invention se rapporte à un système d'information 1 dont l'environnement est de type orienté objet. Le système 1 comporte au moins une machine 2. Le terme « machine » est utilisé dans un sens général pour désigner tout

type d'ordinateurs ou équivalents.

[0021] Le système 1 peut être un système dont l'environnement est de type distribué ou local. Dans la forme de réalisation illustrée sur la figure 1, le système 1 de type distribué comprend trois machines 2a, 2b, 2c, appelées respectivement machine 2a annuaire, machine 2b de stockage, machine 2c d'administration. Seuls les composants des machines 2 du système 1 caractéristiques de la présente invention seront décrits, les autres composants étant connus de l'homme du métier. Les machines 2 sont reliées entre elles au travers d'un réseau 3 ; dans la forme de réalisation de la figure 1, la machine 2a est reliée aux machines 2b et 2c par le réseau 3.

[0022] Le système 1 comporte une base de données ou un annuaire 4 ou équivalent installé sur la machine 2a annuaire. Dans la description qui suit, il sera fait référence à l'annuaire 4. L'annuaire 4 contient des objets 5 caractérisés par des attributs de type connu dits attributs standards 6. Dans l'exemple illustré sur la figure 1, les informations sont structurées en objets 5 intégrés dans l'annuaire 4 suivant le modèle X500 connu ; la machine 2a annuaire communique avec les machines 2b de stockage et 2c d'administration au travers d'une interface 7 LDAP.

[0023] Des utilisateurs ou des applications diverses ou autres, appelés ci-après entités 8 sont susceptibles d'interroger l'annuaire 4 directement sur la machine 2a annuaire l'hébergeant ou à partir d'autres machines 2b, 2c. Dans l'exemple illustré sur la figure 1, une application 8 est installée sur la machine 2c d'administration et travaille avec les données de l'annuaire 4. Selon une forme de réalisation particulière, l'application 8 est munie d'une interface graphique 9.

[0024] L'une des machines 2a, 2b ou 2c du système 1 (machine 2b de stockage sur la figure 1) comprend un compilateur 10 de description et des moyens de stockage 11 d'au moins un fichier 12 source de description.

[0025] L'une des machines 2a, 2b ou 2c du système 1 (machine 2c d'administration sur la figure 1) comprend une interface de programmation au niveau applicatif, appelée couramment interface API 13 munie d'au moins une fonction 14 de gestion et d'au moins une fonction 15 d'accès à l'annuaire 4, des moyens 16 de mémorisation d'une structure 17 de communication entre l'interface API 13 et l'entité 8, une mémoire tampon 18, un module de sécurité 19.

[0026] Le procédé selon la présente invention consiste à caractériser les objets 5 par des attributs spécifiques 20 dits attributs opaques 20, présentant la même structure externe que les attributs standards 6 et dont la structure interne permet de définir et administrer des composants 21 de manière indépendante à l'annuaire 4.

[0027] Les structures des attributs standards 6 sont de trois types : de type texte directement lisible par l'interface 7, de type binaire représentant une unique information d'un type déterminé qui n'est pas directement lisible par l'interface 7, de type spécialisé requérant des

contrôles de syntaxe.

[0028] Le terme « structure externe » signifie que la structure de l'attribut opaque 20 telle que vue par l'annuaire 4 et l'interface 7 est identique à la structure d'un attribut standard 6. Pour l'annuaire 4 et l'interface 7, les attributs opaques 20 présentent une structure externe de type binaire. Un attribut opaque 20 est considéré par l'interface 7 entre l'entité 8 et l'annuaire 4 comme un attribut standard 6.

[0029] L'attribut opaque 20 est comparable à un container : des composants 21 sont introduits dans le container. Les composants 21 sont indépendants de la technologie ou de l'organisation de l'annuaire 4. Il est possible de rajouter des composants 21 dans l'attribut opaque 20 sans modifier le schéma de l'annuaire 4. Les composants 21 introduits dans l'attribut opaque 20 peuvent être de types variés.

[0030] L'objet 5 est identifié de manière unique dans l'annuaire 4 : les attributs opaques 20 sont créés à chaque besoin spécifique d'une entité 8. Le contenu des attributs opaques 20 dépend de l'entité 8 contrairement à celui des attributs standards 6 dépendant de l'annuaire 4 et de son schéma de base.

[0031] L'attribut opaque, de type binaire pour l'interface 7, présente une structure interne déterminée susceptible d'être exploitée par l'interface API 13, structure dans laquelle divers types de composants 21 sont classés.

[0032] Selon une forme de réalisation de l'invention illustrée sur la figure 2, la structure interne de l'attribut opaque 20 est constituée de trois parties :

- un en-tête 22 ;
- une table 23 de caractéristiques des composants 21 ;
- un ensemble 24 de valeurs des composants 21.

[0033] La structure interne de l'attribut opaque est prédéfinie de manière à retrouver les composants 21 dans les informations de type binaire de l'attribut opaque 20 comme vu plus haut.

[0034] L'en-tête 22 fournit des informations générales sur l'attribut opaque 20 concerné telles que :

- une version du formalisme de la structure utilisée pour l'attribut opaque 20 et pour les composants 21 contenus dans l'attribut opaque 20 concerné. La version assure une évolution de la structure de l'objet et permet la compatibilité montante et descendante des structures comme il sera vu plus loin ;
- une longueur en octets des données suivantes de l'en-tête (informations de sécurité, date de dernière mise à jour ...) de manière à retrouver la table 23 dans les données binaires de l'attribut, comme vu plus haut ;
- des informations de sécurité assurant l'intégrité, l'origine et la confidentialité des composants contenus dans l'attribut opaque. Les attributs sont protégés.

gés d'altérations ou d'extractions non souhaitées. Des services de signature et/ou de chiffrement sont susceptibles d'être mis en oeuvre. Les informations de sécurité permettent la signature de l'attribut de manière globale, ce qui améliore les performances. Les informations de sécurité sont par exemple l'indication d'une signature souhaitée ou non pour l'attribut, l'indication d'un chiffrement souhaité pour tous les composants de l'attribut ou seulement certains d'entre eux ou encore non souhaité, la force des algorithmes utilisés, les algorithmes utilisés (signature DSA, RSA... ; chiffrement DES, IDEA...), l'état des composants de l'attribut dans l'ensemble 24 (signé ou non, chiffré ou non), la clé utilisée, la valeur de la signature,...

- une date de dernière mise à jour de l'attribut opaque 20 concerné ; ...

[0035] La table 23 indique les caractéristiques des composants 21 contenus dans l'attribut opaque 20 en question. Les caractéristiques des composants 21 permettent notamment de retrouver un composant 21 déterminé dans l'ensemble 24 des composants 21. La table 23 comprend :

- l'identifiant d'un composant 21 sous la forme par exemple d'un numéro d'identification ;
- un type de codage du composant 21 concerné. Le type de codage décrit le mode de représentation des composants 21 dans l'attribut opaque 20. L'attribut opaque est susceptible de contenir des composants de types variés. Le type permet le contrôle du stockage des composants 21 dans l'attribut opaque 20, l'organisation des composants 21 en liste ;
- des informations complémentaires de codage comme par exemple des informations concernant le type multimédia ;
- une longueur utile de valeur du composant 21 concerné exprimée en octets ; ...

[0036] Les composants 21 sont stockés dans l'ensemble 24 les uns à la suite des autres dans l'ordre dans lequel ils ont été cités dans la table 23. L'identifiant et la longueur utile indiqués dans la table 23 permettent de retrouver un composant 21 déterminé dans l'ensemble 24.

[0037] L'attribut opaque 20 permet de regrouper des composants 21 de types différents et de permettre l'ajout, la modification ou la suppression d'un attribut 6, 20 et/ou d'un composant 21 sans intervenir sur le schéma de l'annuaire 4.

[0038] Selon une forme de réalisation, les types de codage déterminés dans la table 23 se divisent en trois catégories : les types simples, les types multimédia, les types constructeurs.

[0039] Les types simples comprennent le type texte (« TEXT » en annexe) qui correspond à une chaîne de caractères (texte quelconque), le type entier de valeur

binaire, le type numérique (caractères numériques de 0 à 9), le type bit qui correspond à un champ d'octets, le type booléen (valeur vrai ou faux), le type énuméré dont la valeur est associée à une constante, le type secret (texte spécial de type chiffré et de longueur quelconque), le type date/heure (format de date/heure établi).

[0040] Les types multimédia comportent le type image, le type son et le type vidéo.

[0041] Les types constructeurs introduisent et délimitent des structures telles que des listes. Le type liste (« LIST » en annexe) représente une liste de n occurrences d'une structure, le nombre maximum d'occurrences étant fixé. Le type fin de liste (« END » en annexe) clôt la description d'une liste. Les types constructeurs autorisent l'imbrication des listes ; n niveaux d'imbrication sont possibles. Des règles d'utilisation accompagnent les types constructeurs.

[0042] Tout autre type est susceptible d'être utilisé.

[0043] Les fonctions 15 d'accès permettent l'accès aux attributs 6, 20 de l'annuaire 4. Les fonctions 15 d'accès permettent d'extraire, d'introduire, de modifier ou de supprimer des attributs dans l'annuaire 4. Les fonctions 15 mettent les attributs extraits de l'annuaire 4 à la disposition des fonctions 14 de gestion dans la mémoire tampon 18. Les fonctions 14 de gestion créent, modifient, suppriment des attributs et/ou des composants et transmettent le résultat des opérations aux fonctions 15 d'accès par l'intermédiaire de la mémoire tampon 18 : les fonctions 15 d'accès introduisent les attributs et/ou composants créés ou modifiés par les fonctions 14 dans l'annuaire 4, ou encore suppriment de l'annuaire 4 les attributs et/ou composants supprimés par les fonctions 14 de gestion.

[0044] Les fonctions 14 de gestion sont prévues pour l'exploitation des attributs opaques 20 de manière indépendante à l'annuaire 4.

[0045] Les fonctions 14 de gestion permettent d'introduire ou de supprimer des attributs standards 6 dans des objets 5, des composants 21 dans des attributs opaques 20 ou des attributs opaques 20 dans des objets, de lire ou modifier des attributs standards 6 ou des composants 21 des attributs opaques 20, en s'appuyant si nécessaire sur une description 25 formelle compilée et stockée dans l'annuaire 4. Les fonctions 14 de gestion sont susceptibles de lire les composants 21 des attributs opaques 20 sans faire appel à la description 25 correspondante : les fonctions 14 de gestion doivent alors connaître l'identifiant du composant recherché (l'identifiant d'un attribut opaque est donné par la table 23 dudit attribut). L'attribut opaque contient le minimum d'informations permettant aux fonctions de gestion 14 de lire les composants qu'il contient sans informations supplémentaires. Il est à noter que la lecture est l'opération la plus fréquente statistiquement. Pour toute autre opération telle la création, la modification ou la suppression d'un attribut 6, 20 ou d'un composant 21, la fonction 14 de gestion requiert les informations de la description 25 compilée. Lors d'opérations de mise à jour (création,

modification ou suppression), des contrôles sont nécessaires tels que des contrôles sur les longueurs, les types, les masques, les caractères obligatoires... : les contrôles s'appuient sur les informations comprises dans les descriptions 25 compilées.

[0046] Il est à noter que les informations contenues dans les descriptions 25 pourraient, selon une forme de réalisation de l'invention, être comprises dans les attributs opaques 20. Les descriptions 25 compilées améliorent les performances du système 1.

[0047] La description 25 de l'attribut opaque 20 est issue des fichiers source 12. Un fichier source 12 est réservé pour chaque description 25 et pour chaque version de la description 25. La compilation de la description 25 s'effectue à partir des fichiers source 12 au moyen du compilateur 10. La compilation améliore les performances de traitement.

[0048] La description 25 compilée apporte à l'entité 8 des informations relatives à un attribut opaque donné sans que l'entité ait à solliciter l'annuaire 4: le fichier source 12 contient des informations telles que le libellé externe de saisie du champ attribut ou composant « LBL » comme il sera vu plus loin, assurant une mise en forme automatique des formulaires de saisie de l'interface graphique 9 de l'entité 8: l'interface 9 est générée automatiquement sans avoir recours aux attributs stockés dans l'annuaire 4. La même description 25 est utilisée pour la saisie de données extérieures au système 1 et la gestion de données appartenant à l'annuaire 4, ce qui assure une cohérence dans la gestion des informations.

[0049] Dans la forme de réalisation des figures 1 et 3, les fichiers source 12 sont enregistrés dans la machine 2b de stockage. Le compilateur 10 de la machine 2b transforme le fichier source contenu dans les moyens de stockage 11 en une description 25 compilée stockée dans l'annuaire 4 de la machine 2a annuaire. La description compilée liée à un objet 5 donné est stockée par exemple (figure 1) dans un objet 26 spécial. Le fichier compilé bénéficie des capacités de distribution des technologies d'annuaires.

[0050] La description 25 compilée permet une évolution en douceur du contenu des objets. Selon un exemple de forme de réalisation, la description 25 comporte, comme il sera vu plus loin, le numéro de version du contenu de l'objet 5 concerné : l'évolution de structure des objets 5 est gérée à l'aide de numéros croissants de version.

[0051] La gestion des versions est rendue souple et efficace. L'annuaire 4 peut évoluer sans cesse sans remettre en cause les objets existants et le schéma de l'annuaire. La gestion des versions est assurée par les fonctions 14 de gestion lors de la création ou suppression d'objets 5 dans l'annuaire 4.

[0052] Selon un exemple de réalisation de la gestion des versions, il est prévu une compatibilité montante entre la description 25 d'un objet 5 et l'objet 5 effectif existant (objet 5 de version V et description 25 de version V+n):

[0053] Dans ce cas, afin d'assurer une totale compatibilité, c'est à dire une évolution de la structure des objets 5 sans remettre en cause les objets existants, les règles suivantes sont appliquées dans les fonctions de gestion 14:

- Si un nouveau composant 21 apparaît dans la description 25, il est visible pour l'entité 8 et indiqué comme absent dans l'objet 5 courant mais ajouté en mise à jour par la fonction de gestion 14 concernée ;
- Si un composant 21 disparaît de la description 25, il n'est plus visible pour l'entité 8 et est supprimé de l'objet 5 à la mise à jour par la fonction de gestion 14 concernée.

[0054] Il est également prévu une compatibilité descendante entre la description d'un objet et l'objet effectif existant (objet de version V et description de version V-n) :

[0055] Cette situation intervient dans des périodes transitoires de déploiement d'une nouvelle version de description d'objet, et de différences dues à la réplication des objets.

[0056] Dans ce cas, une compatibilité minimum est assurée, permettant un fonctionnement transitoire, en attendant la mise à niveau de la description.

[0057] Les règles suivantes sont appliquées :

- Si un composant 21 est présent dans la description 25 mais ne l'est plus dans l'objet 5, le composant 21 est considéré comme absent de la description ;
- Si un composant est absent de la description mais présent dans l'objet, il est considéré comme présent dans la description, accessible dans l'annuaire 4 au moyen de l'identifiant mais pas du nom interne qui n'est défini que dans la description.

[0058] Les modifications à apporter au schéma de l'annuaire 4 sont limitées : les évolutions sont déportées hors de l'annuaire 4.

[0059] Dans la forme de réalisation de la figure 1, les attributs d'une classe d'objets 5 sont regroupés en items. Une classe d'objets peut contenir plusieurs items. Dans la forme de réalisation décrite, un item est constitué d'au moins un attribut standard 6 et/ou d'un seul attribut opaque 20. Un attribut standard 6 ou un composant 21 dans un item est appelé dans ce qui suit un élément de l'item.

[0060] D'autres items sont susceptibles d'être créés.

[0061] Les descriptions 25 sont de deux types :

- les descriptions 25a d'items ;
- les descriptions 25b d'attributs.

[0062] L'annexe 1 montre un exemple de description 25 d'items et d'attributs.

[0063] La description 25a d'items est référencée sous

la forme <Classe> - <Item>, <Classe> représente la classe de l'objet 5 concerné, <Item> l'item concerné (ensemble de composants d'un attribut opaque ou ensemble d'attributs standards dans ladite classe d'objets en question). Les items cités sont déclarés au préalable comme appartenant à la classe d'objets correspondante dans l'annuaire 4. Un fichier source 12 peut comporter une description d'items et une suite de descriptions d'attributs correspondants.

[0064] La description 25 d'items comprend les informations suivantes :

- la classe de l'objet concerné, appelée « CLASS ». Les classes sont identifiées par des mots clés réservés.
- les items déclarés dans un objet qui dépendent de la classe de l'objet en question, appelés « ITEM ». Une classe d'objets est constituée de plusieurs items. Pour chaque classe, la liste des items qu'il doit/peut contenir est définie. A chaque type d'item est associé un identifiant unique. Dans la forme de réalisation décrite, les identifiants sont :
 - « BASE » : attributs standards 6 ;
 - « XXX » (tout terme différent du terme BASE utilisé pour les attributs standards) : attributs opaques 20.
- la version de la description de l'item, appelée « VERSION ». Cette information est très importante : elle assure une évolution compatible de la description des items.
- la liste des langues à utiliser, appelée « LANGUAGES » telles que par exemple la langue utilisée pour les libellés d'attributs, ... ; une description requiert au moins une langue. Si une des langues n'est pas fournie explicitement pour un attribut, on prend par défaut la première langue définie dans la description pour l'attribut en question.

[0065] La description 25 d'items peut également comprendre des informations de sécurité optionnelles. Les informations sont dites optionnelles car elles impliquent une option par défaut en cas d'absence. Les informations de sécurité sont définies dans l'en-tête 22 de l'attribut opaque 20 et/ou dans la description 25a d'items selon les besoins (options de sécurité générales et/ou pour seulement certains attributs). La protection des attributs opaques 20 s'effectue par chiffrement et/ou signature. Par défaut, par exemple, les attributs opaques ne sont ni signés ni chiffrés. Les paramètres optionnels de la description 25a autorisent la modification des valeurs de protection par défaut.

[0066] Selon une forme de réalisation, les paramètres optionnels sont :

- la signature de l'attribut en question ;
- le chiffrement de l'attribut concerné ; dans le cas d'un chiffrement requis, les composants 21 de type secret ne sont pas chiffrés individuellement.
- le niveau de chiffrement pour l'attribut en question.

[0067] La description 25b d'attributs permet :

- d'identifier chaque élément de l'objet concerné par un nom interne et un identifiant, d'indiquer la correspondance entre le nom interne et l'identifiant pour un élément donné ;
- de définir les types des éléments de l'objet concerné ;
- de fournir des informations utilisables par les fonctions de gestion 14 des éléments 21 ou directement par l'entité 8 (interface graphique 9) telles que :
 - un libellé(s) du champ de saisie de l'élément ;
 - une indication du statut obligatoire ou optionnel de l'élément ;
 - une taille minimum / maximum ;
 - des contraintes sur l'élément ;
 - des informations complémentaires sur l'élément.

[0068] La description comprend :

- le nom interne de l'élément concerné, appelé « ATT ». Le nom interne permet de nommer simplement un élément (attribut standard 6 / composant 21 d'un attribut opaque 20). Il sert de référence pour la gestion ultérieure des descriptions d'items et pour l'administration de l'annuaire 4. Le nom interne est indépendant du nom défini dans le schéma de l'annuaire 4. Le nom interne identifie chaque élément ; il est unique pour un objet 5 donné. Selon un exemple illustratif, le nom est donné sur 12 caractères maximum. Le nom doit commencer sur le premier caractère de la ligne ;
- l'identifiant, appelé l'OID. L'identifiant identifie chaque élément ; il est unique pour un item donné. L'identifiant est indépendant de l'identification dans l'annuaire 4.
- le nom de l'élément dans l'annuaire 4 utilisé par l'interface 7 et obligatoire dans la description des attributs 6 standards, appelé « XNM ». Il est codé par exemple sur 36 caractères maximum ;
- l'indication de l'état de l'élément au niveau de l'entité 8, appelé « HID ». L'état de l'élément peut prendre les valeurs cachés ou visibles. L'HID est déclaré avant les options qui suivent ;
- le libellé externe de saisie du champ élément tel que présenté au niveau de l'entité 8, appelé « LBL » ;

- le type de codage de l'élément parmi ceux énoncés précédemment, appelé « TYP ».

[0069] D'autres options sont susceptibles d'être utilisées telles que par exemple des options qui sont fonction du type de codage de l'élément. Quelques exemples sont donnés ci-après permettant de comprendre l'exemple de fichiers source donné en annexe.

- l'option « MDT » détermine le statut obligatoire ou optionnel de l'élément ;
- l'option « RNG » représente la plage de longueur (chaîne de caractères, numérique, secret) ou de valeur (entier), ou le nombre d'occurrences minimal et maximal (type constructeur).
- l'option « SZF » définit la taille du champ de saisie pour l'interface graphique 9 de l'entité 8.
- « CAS » définit la casse pour les types chaînes de caractères ;

« FREE », la casse est respectée ;
 « UP », les majuscules sont forcées ;
 « LOW », les minuscules sont forcées.

[0070] Les informations données par la description telles que l'option SZF par exemple permettent la mise en forme automatique (sans avoir recours à des modules supplémentaires) des formulaires de saisie de l'interface graphique.

[0071] Des options sont à prévoir également par exemple pour les types multimédia tels que :

- l'option « PLD » indiquant le préchargement automatique des types multimédias dans la structure.
- l'option « ALG » qui est un identifiant d'algorithme.

[0072] Les identifiants d'algorithme pour le type image sont par exemple :

0	GIF
1	JPG
2	TIF (fractales)
3	Ondelettes

Des identifiants sont également déterminés pour les types sons et vidéo.

[0073] La manipulation des éléments s'effectue à l'aide des fonctions de gestion 14 en précisant l'item. L'item doit être précisé à l'appel afin que les fonctions déterminent la structure concernée par l'opération à effectuer.

[0074] Les fonctions de gestion 14 permettent de manipuler (créer, lire, modifier, supprimer, afficher) les données suivantes :

- attributs standards des objets de l'annuaire 4 ;
- attributs opaques des objets de l'annuaire 4 et composants 21 de ces attributs ;
- composants de sécurité de base intégrés dans des attributs des objets 5...

[0075] Les éléments manipulés par les entités 8 sont contrôlés, filtrés par les fonctions 14 de gestion.

[0076] Si la fonction de gestion 14 appelée nécessite les descriptions 25a et b compilées de l'objet, lesdites descriptions sont chargées automatiquement par la fonction de gestion en question. Selon une forme de réalisation particulière de réalisation, les descriptions compilées sont chargées dans la mémoire tampon 18. Les fonctions de gestion 14 puisent dans les descriptions compilées les informations nécessaires aux opérations qu'elles doivent effectuer. Les fonctions de gestion 14 s'appuient sur la description 25a d'item pour connaître la classe d'objet et l'item concernés.

[0077] Les fonctions de gestion dans la forme de réalisation décrite sont les suivantes :

- une fonction d'initialisation d'objets InitObj : l'appel à cette fonction permet d'initialiser la structure d'un nouvel objet ainsi que la structure d'attributs 6,20 de l'objet créé de manière à pouvoir y insérer des attributs standards 6 ou des composants 21. L'objet 5 est défini à l'appel de la fonction d'initialisation en précisant la classe de l'objet, l'item correspondant.
- une fonction de rangement PutElem : la fonction de rangement réalise la création ou mise à jour d'un attribut 6,20 ou d'un composant 21 d'un attribut opaque 20. A l'appel de la fonction de rangement sont précisés la version, la classe, l'identifiant ou le nom de l'élément, l'occurrence en cas de liste, la longueur utile dans le cas de types bit ou secret, le type de codage de l'élément. La fonction de rangement actualise pour chaque objet traité une liste des attributs modifiées afin d'optimiser la mise à jour.
- une fonction de lecture GetElem : la fonction de lecture assure l'extraction d'un attribut 6, 20 ou d'un composant 21 d'un attribut opaque 20 de l'annuaire 4. Si le composant demandé est recherché par son identifiant d'objets, la fonction de lecture n'a pas à charger la description compilée : la table 23 intégrée de l'attribut est suffisante. A l'appel de la fonction de lecture, sont précisés la version, la classe, l'identifiant ou le nom de l'élément, l'occurrence en cas de liste, la longueur utile dans le cas de types bit ou secret, le type de l'élément.
- une fonction de suppression RemElem : la fonction de suppression effectue la suppression de la valeur d'un attribut 6,20 ou d'un composant 21 d'un attribut opaque 20, d'une occurrence d'une liste de composants dans un objet. A l'appel de la fonction de sup-

pression, sont précisés la version, la classe, l'identifiant ou le nom de l'élément, l'occurrence en cas de liste, la longueur utile dans le cas de types bit ou secret, le type de codage de l'élément.

- une fonction de vérification CheckObj : la fonction de vérification vérifie les attributs 6,20 d'un objet et les composants 21 d'un attribut opaque 20 avant validation d'une création ou d'une modification. La fonction de vérification permet le contrôle de présence obligatoire des attributs/composants. Elle peut être appelée par une entité 8, mais elle peut également être mise en oeuvre systématiquement. A l'appel de la fonction de vérification, sont précisés la version, la classe, l'identifiant ou le nom de l'élément, l'occurrence en cas de liste, la longueur utile dans le cas de types bit ou secret, le type de codage de l'élément.

[0078] L'entité 8 appelle une fonction 14 de gestion en plaçant les paramètres nécessaires à ladite fonction 14 dans la structure de communication 17. La structure de communication 17 se présente sous la forme d'une succession de tous les paramètres nécessaires à toutes les fonctions 14 de gestion. La fonction 14 de gestion appelée récupère dans la structure 17 de communication les valeurs de paramètres qui l'intéressent.

[0079] Les fonctions 14 de gestion renvoient un code retour indiquant si la fonction a été correctement effectuée, si une erreur est intervenue, si les paramètres d'appel sont corrects, si la description compilée a été retrouvée, si l'objet existe ...

[0080] Les fonctions 14 de gestion sont conçues de manière à suivre les règles de gestion de versions énoncées plus haut.

[0081] Les fonctions 14 de gestion permettent de lire une suite d'attributs/ composants sans solliciter l'annuaire 4.

[0082] Les fonctions 14 de gestion assurent le chiffrement/déchiffrement et la signature des composants 21 contenus dans l'attribut opaque 20. Les fonctions de gestion communiquent avec le module de sécurité 19 pour effectuer les opérations de sécurité (signature, chiffrement, ...) requises dans une description d'item 25 et/ou dans l'en-tête 22 d'un attribut opaque. Il est à noter que le module de sécurité 19 est indépendant : l'entité 8 peut faire appel directement au module de sécurité comme le montre la figure 1.

[0083] Le système 1 décrit fonctionne de la manière suivante :

[0084] Lorsque l'entité 8 souhaite créer un objet dans l'annuaire 4, l'entité 8 commence par appeler la fonction de gestion 14 d'initialisation : l'entité 8 introduit les paramètres d'appel suffisants dans la structure de communication 17. La fonction 14 d'initialisation extrait de ladite structure les paramètres qui lui sont nécessaires, charge la description 25 compilée correspondante dans la mémoire tampon 18 et crée la structure de départ

d'objet souhaitée en définissant des valeurs par défaut. La structure est enregistrée dans la mémoire tampon 18. La structure créée, la fonction 14 d'initialisation indique à l'entité 8 que la structure est prête à recevoir des attributs standards 6 ou des composants 21 en renvoyant un code retour adapté dans la structure de communication 17. Comme indiqué plus haut, divers codes retour sont susceptibles d'être transmis à l'entité 8 pour lui indiquer l'état de fonctionnement. Lorsque l'entité détecte le code retour indiquant que la structure de départ a été créée, l'entité 8 appelle alors la fonction 14 de gestion de rangement : elle transmet à la structure de communication les caractéristiques des attributs standards 6 ou des composants 21 qu'elle souhaite introduire dans l'attribut opaque 20 concerné. La fonction 14 de rangement les récupère et remplit la structure d'objets précédemment créée dans la mémoire tampon 18 en s'appuyant sur la description compilée chargée. La fonction 14 de rangement peut également faire appel au module 19 de sécurité en cas de signature ou de chiffrement des informations contenues dans l'objet concerné. Lorsque l'objet est prêt à être transféré dans l'annuaire 4, l'entité 8 a le choix entre valider la création ou la vérifier sachant qu'en validant, l'opération de vérification est également réalisée. La vérification avant validation permet d'obtenir précisément toutes les erreurs réalisées. L'opération de vérification est effectuée par la fonction de gestion de vérification. Lorsque l'entité 8 valide la création, la fonction 15 d'accès transfère l'objet créé et renseigné dans l'annuaire 4.

[0085] Lorsque l'entité 8 souhaite modifier un objet dans l'annuaire 4, l'entité 8 commence par appeler la fonction 15 d'accès. La fonction 15 d'accès retrouve l'objet dans l'annuaire 4 et en transfère une copie dans la mémoire tampon 18. La fonction 15 d'accès indique par code retour à l'entité 8 que l'objet a été correctement transféré dans la mémoire tampon 18. L'entité 8 appelle alors la fonction 14 de gestion de rangement : elle transmet à la structure de communication 17 les caractéristiques modifiées de l'attribut standard 6 ou du composant 21 de l'attribut opaque 20 concerné. La fonction 14 de rangement extrait les paramètres qui lui sont nécessaires de la structure de communication 17 et charge la description compilée correspondante dans la mémoire tampon 18. La fonction 14 de rangement modifie l'objet en question enregistré dans la mémoire tampon 18 par la fonction d'accès 15, comme indiqué par les paramètres de la structure de communication transmis par l'entité 8 et conformément à la description compilée. La fonction de rangement peut également faire appel au module de sécurité 19 en cas de signature ou de chiffrement des informations contenues dans l'objet concerné. Lorsque l'objet est prêt à être transféré dans l'annuaire 4, l'entité 8 a le choix entre valider la création ou la vérifier sachant qu'en validant, l'opération de vérification est également réalisée. La vérification avant validation permet d'obtenir précisément toutes les erreurs réalisées. L'opération de vérification est effectuée par la

fonction de gestion de vérification. Lorsque l'entité 8 valide la modification, la fonction 15 d'accès transfère l'objet modifié de la mémoire tampon 18 dans l'annuaire 4.

[0086] Lorsque l'entité 8 souhaite supprimer un attribut 6, 20 ou un composant 21 de l'annuaire 4, l'entité 8 commence par appeler la fonction 15 d'accès. La fonction 15 d'accès retrouve l'objet dans l'annuaire 4 et en transfère une copie dans la mémoire tampon 18. La fonction 15 d'accès indique par code retour à l'entité 8 que l'objet a été correctement transféré dans la mémoire tampon 18. L'entité 8 appelle alors la fonction 14 de gestion de suppression : elle transmet à la structure de communication les caractéristiques de l'attribut standard 6 ou du composant 21 à supprimer de l'attribut opaque 20 concerné. La fonction 14 de suppression extrait les paramètres qui lui sont nécessaires de la structure de communication 17 et charge la description compilée correspondante dans la mémoire tampon 18. La fonction 14 de suppression supprime le composant 21 ou l'attribut 6,20 en question enregistré dans la mémoire tampon 18 par la fonction d'accès 15, comme indiqué par les paramètres de la structure de communication transmis par l'entité 8 et conformément à la description compilée. La fonction 15 d'accès supprime l'attribut 6, 20 ou le composant 21 indiqué dans la mémoire tampon 18 dans l'annuaire 4.

[0087] Lorsque l'entité 8 souhaite lire un objet dans l'annuaire 4, l'entité 8 commence par appeler la fonction 15 d'accès. La fonction 15 d'accès retrouve l'objet dans l'annuaire 4 et en transfère une copie dans la mémoire tampon 18. La fonction d'accès indique par code retour à l'entité 8 que l'objet a été correctement transféré dans la mémoire tampon 18. L'entité 8 appelle alors la fonction de gestion 14 de lecture. Si les paramètres fournis par l'entité 8 sont suffisants et en particulier si le nom interne de l'attribut standard ou du composant a été donné, la fonction de lecture n'a pas besoin de charger la description compilée. Dans le cas contraire, la description compilée est chargée de manière à obtenir la correspondance entre le nom interne et l'identifiant. La fonction 14 de lecture vérifie éventuellement si l'entité 8 a le droit de lire les informations concernées en faisant appel au module de sécurité 19. Elle déchiffre éventuellement lesdites informations et vérifie leur intégrité. Elle retrouve les informations souhaitées de l'objet en parcourant l'ensemble 24 des valeurs à l'aide de l'identifiant ou libellé et de la longueur utile. Elle place lesdites informations dans la structure de communication 17. L'entité 8 n'a plus qu'à lire les informations requises dans la structure de communication 17.

[0088] Selon un développement de l'invention, les versions de description antérieures à une version utilisée à un instant t sont archivées dans un répertoire séparé. Les versions de description antérieures peuvent être archivées par exemple en ajoutant au nom un suffixe représentant le numéro de version.

[0089] La présente invention concerne donc un procédé de gestion de l'annuaire 4 ou équivalent compor-

tant les objets 5 caractérisés par les attributs standards 6, l'annuaire 4 étant apte à être interrogé par l'entité 8, caractérisé en ce qu'il consiste à caractériser les objets 5 par les attributs spécifiques 20 dits attributs opaques 20, présentant la même structure externe que les attributs standards 6 et dont la structure interne permet de définir et administrer les composants 21 de manière indépendante à l'annuaire 4.

[0090] Le procédé selon l'invention consiste à définir et administrer les composants 21 au moyen de l'interface de programmation au niveau applicatif API 13 entre l'annuaire 4 et l'entité 8.

[0091] Le procédé consiste à définir la structure interne de l'attribut opaque en trois parties :

- l'en-tête 22 délivrant des informations générales sur l'attribut opaque 20 ;
- la table 23 de caractéristiques des composants 21 ;
- l'ensemble 24 de valeurs des composants 21 indiqués dans la table 23.

[0092] Le procédé consiste à intégrer dans la structure interne de l'attribut opaque 20 des informations de sécurité assurant l'intégrité, l'origine et/ou la confidentialité des composants 21 contenus dans l'attribut 20.

[0093] Le procédé consiste à intégrer dans la structure interne de l'attribut opaque 20 des informations permettant la gestion de différentes versions du contenu de l'attribut 20.

[0094] Le procédé consiste à intégrer dans la structure interne de l'attribut opaque 20 le type de codage de chaque composant 21 contenu dans ledit attribut 20.

[0095] Le procédé consiste à lire ou mettre à jour, à savoir créer, modifier, lire ou supprimer, un attribut standard 6, un attribut opaque ou un composant 21 dans un attribut opaque 20 de manière indépendante à l'annuaire 4 au moyen des fonctions de gestion 14 des attributs opaques 20 appelées par l'entité 8.

[0096] Les fonctions 14 de gestion s'appuient sur la description 25 des attributs 6,20 pour effectuer leur mise à jour, ainsi que leur lecture si l'entité 8 n'a pas fourni les informations suffisantes pour accéder directement à l'attribut concerné dans l'annuaire 4.

[0097] Le procédé consiste à compiler la description 25 à l'aide d'un compilateur 10.

[0098] Le procédé consiste à intégrer dans la description 25 des informations de sécurité assurant l'intégrité, l'origine et/ou la confidentialité des composants 21 contenus dans l'attribut 20.

[0099] Le procédé consiste, au moyen des fonctions 15 d'accès à l'annuaire 4, à extraire, introduire, modifier ou supprimer des attributs 6,20 dans l'annuaire 4 conformément aux opérations effectuées par les fonctions 14 de gestion.

[0100] La présente invention porte également sur le système informatique permettant la mise en oeuvre dudit procédé.

[0101] La présente invention se rapporte à un systè-

me informatique comprenant au moins une machine 2 munie de l'annuaire 4 ou équivalent susceptible d'être interrogé par l'entité 8, caractérisé en ce qu'il comprend l'interface de programmation au niveau applicatif API 13 entre l'annuaire 4 et l'entité 8 comprenant les fonctions 15 d'accès à l'annuaire 4 et les fonctions 14 de gestion d'attributs standards 6 et/ou d'attributs opaques 20.

[0102] L'entité 8 communique avec l'interface API 13 au travers de la structure de communication 17 enregistrée dans les moyens de mémorisation 16.

[0103] Le système comprend les moyens 11 de stockage des fichiers source 12, le compilateur 10 desdits fichiers 12.

ANNEXES

Exemples de fichier source 12.

[0104] / description 25 de l'item BASE de la classe d'objets Personne

CLASS=PERSONNE
ITEM=BASE
VERSION=001
LANGUAGES=EN,FR

/ description 21 des attributs standards 6 de l'item BASE

ATT=CNAME
OID=0001
XNM=commonName
HID=N
LBL=EN="User common name"
LBL=FR="ident X500 de l'utilisateur"
TYP=TEXT
MDT=Y
RNG=64
SZF=24
CAS=FREE

ATT=SURNAME
OID=0002
XNM=surname
HID=N
LBL=EN="User surname"
LBL=FR="Patronyme X500 de l'utilisateur"
TYP=TEXT
MDT=Y
RNG=32
SZF=16
CAS=LOW

ATT=TELEPHONE
OID=0003
XNM= telephoneNumber
HID=N
LBL=EN="User telephone"

LBL=FR="Telephone usager"
MDT=N
RNG=16
SZF=16
CAS=UP

/ description 25 d'un item attribut opaque SECITEM 20 de la classe d'objets PERSON

/ description des composants de l'attribut opaque, dans l'item SECITEM

CLASS=PERSON
ITEM=SECITEM
VERSION=001
LANGUAGES=EN,FR

/ description 25 des attributs opaques 20 de l'item / définition de la liste des ressources accordées

20 ATT=LISTRAL
OID=100
LBL=EN="Resource allocated list"
LBL=FR="Liste des ressources allouées"
TYP=LIST
25 PRS=PCM
RNG=500

/ classe des ressources de la liste

30 ATT=OBJECTCLASS
OID=101
LBL=EN="Resource class"
LBL=FR="Classe de la ressource"
TYP=TEXT
35 RNG=48
CAS=UP

/ nom de la ressource

40 ATT=OBJECTID
OID=102
LBL=EN="Resource name"
LBL=FR="Nom de la ressource"
TYP=TEXT
45 MDT=N
SZF=48
RNG=96
CAS=UP

50 / fin description de la liste (d'une occurrence liste)

ATT=ENDOFLLIST
OID=100
HID=Y
55 TYP=END

/ exemple de description des droits d'accès aux noeuds et objets

/ droit sur le niveau noeud

ATT=NODERGT

OID=102

HID=N

LBL=EN="Rights on node"

LBL=FR="Droits sur le noeud"

TYP=TEXT

MDT=N

RNG=8

CAS=UP

ATT=CHILDGRGT

OID=103

HID=N

LBL=EN="General rights on attached objects"

LBL=FR="Droits généraux sur les objets rattachées"

TYP=TEXT

MDT=N

RNG=8

CAS=UP

ATT=CLASSRGT

OID=104

HID=N

LBL=EN="Specific rights on attached objects"

LBL=FR="Droits spécifiques sur les objets rattachées"

TYP=TEXT

MDT=N

RNG=8

CAS=UP

Revendications

1. Procédé de gestion d'une base de données (4) ou annuaire ou équivalent comportant des objets (5) caractérisés par des attributs standards (6), la base (4) étant apte à être interrogée par une application, un utilisateur ou autre appelé entité (8), caractérisé en ce qu'il consiste à caractériser les objets (5) par des attributs spécifiques (20) dits attributs opaques (20), présentant la même structure externe que les attributs standards (6) et dont la structure interne permet de définir et administrer des composants (21) de manière indépendante à la base (4). 40
2. Procédé selon la revendication 1, caractérisé en ce qu'il consiste à définir et administrer les composants (21) au moyen d'une interface de programmation au niveau applicatif API (13) entre la base (4) et l'entité (8). 50
3. Procédé selon l'une des revendications 1 ou 2, caractérisé en ce qu'il consiste à définir la structure interne de l'attribut opaque en trois parties : 55

- un en-tête (22) délivrant des informations générales sur l'attribut opaque (20);
- une table (23) de caractéristiques des composants (21);
- un ensemble (24) de valeurs des composants (21) indiqués dans la table (23). 5

4. Procédé selon l'une des revendications 1 à 3, caractérisé en ce qu'il consiste à intégrer dans la structure interne de l'attribut opaque (20) des informations de sécurité assurant l'intégrité, l'origine et/ou la confidentialité des composants (21) contenus dans l'attribut (20). 10

5. Procédé selon l'une des revendications 1 à 4, caractérisé en ce qu'il consiste à intégrer dans la structure interne de l'attribut opaque (20) des informations permettant la gestion de différentes versions du contenu de l'attribut (20). 15

6. Procédé selon l'une des revendications 1 à 5, caractérisé en ce qu'il consiste à intégrer dans la structure interne de l'attribut opaque (20) le type de codage de chaque composant (21) contenu dans ledit attribut (20). 20

7. Procédé selon l'une des revendications 1 à 6, caractérisé en ce qu'il consiste à lire ou mettre à jour, à savoir créer, modifier, lire ou supprimer, un attribut standard (6), un attribut opaque ou un composant (21) dans un attribut opaque (20) de manière indépendante à la base (4) au moyen de fonctions de gestion (14) des attributs opaques (20) appelées par l'entité (8). 25

8. Procédé selon la revendication 7, caractérisé en ce que les fonctions (14) de gestion s'appuient sur une description (25) des attributs (6,20) pour effectuer leur mise à jour, ainsi que leur lecture si l'entité (8) n'a pas fourni les informations suffisantes pour accéder directement à l'attribut concerné dans la base (4). 30

9. Procédé selon la revendication 8, caractérisé en ce qu'il consiste à compiler la description (25) à l'aide d'un compilateur (10). 35

10. Procédé selon l'une des revendications 8 ou 9, caractérisé en ce qu'il consiste à intégrer dans la description (25) des informations de sécurité assurant l'intégrité, l'origine et/ou la confidentialité des composants (21) contenus dans l'attribut (20). 40

11. Procédé selon la revendication 7, caractérisé en ce qu'il consiste, au moyen de fonctions (15) d'accès à la base (4), à extraire, introduire, modifier ou supprimer des attributs (6,20) dans la base (4) conformément aux opérations effectuées par les fonctions 45

(14) de gestion.

12. Système informatique comprenant au moins une machine (2) munie d'une base de données (4) ou annuaire ou équivalent susceptible d'être interrogée par une application, un utilisateur ou autre, appelé entité (8), caractérisé en ce qu'il comprend une interface de programmation au niveau applicatif API (13) entre la base (4) et l'entité (8) comprenant des fonctions (15) d'accès à la base (4) et des fonctions (14) de gestion d'attributs standards (6) et/ou d'attributs opaques (20). 5 10
13. Système selon la revendication 12, caractérisé en ce que l'entité (8) communique avec l'interface API (13) au travers d'une structure de communication (17) enregistrée dans des moyens de mémorisation (16). 15
14. Système selon l'une des revendications 12 ou 13, caractérisé en ce qu'il comprend des moyens (11) de stockage de fichiers source (12), un compilateur (10) desdits fichiers (12). 20
15. Système informatique permettant la mise en oeuvre du procédé selon l'une des revendications 1 à 11. 25

30

35

40

45

50

55

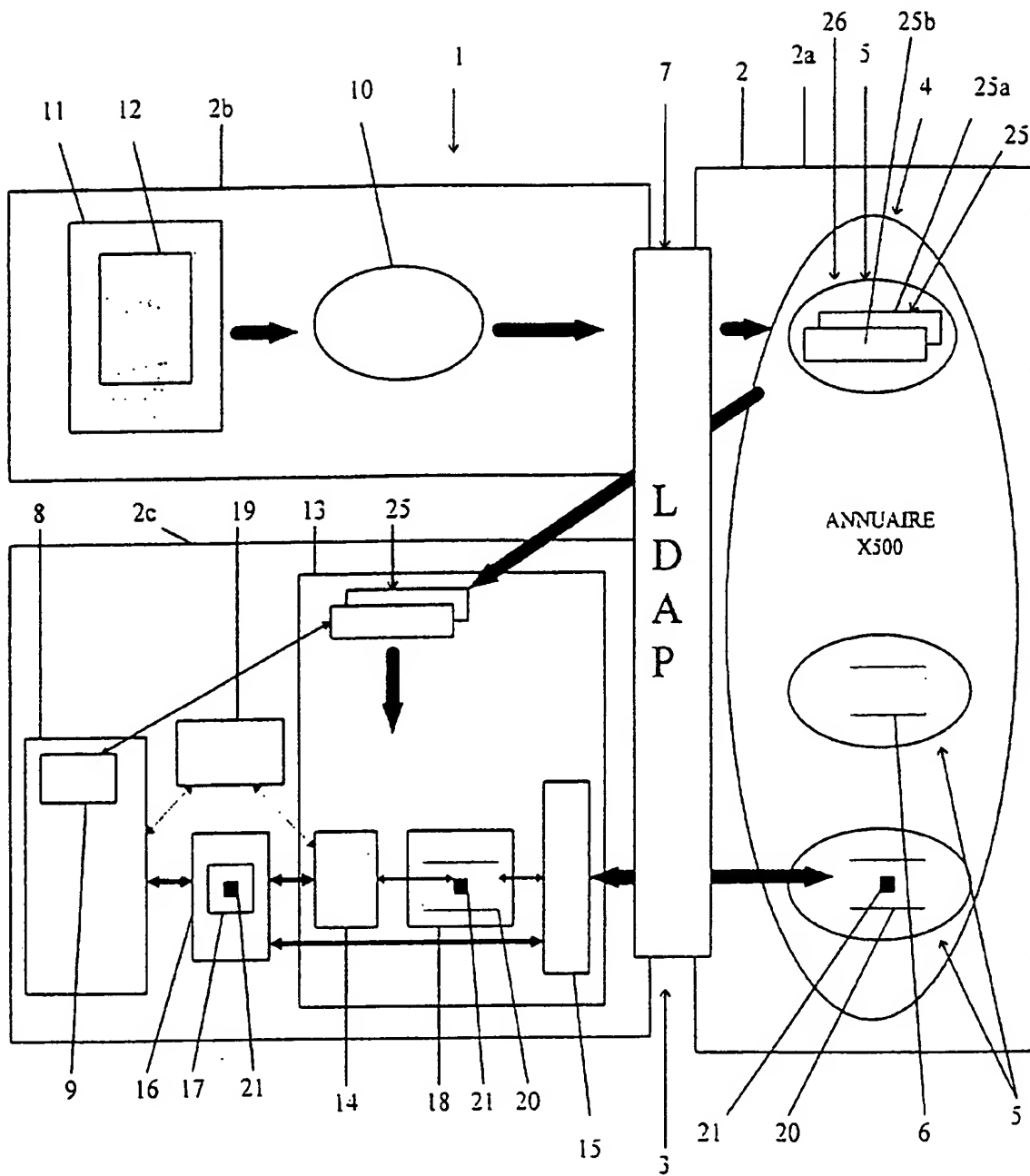


FIG.1

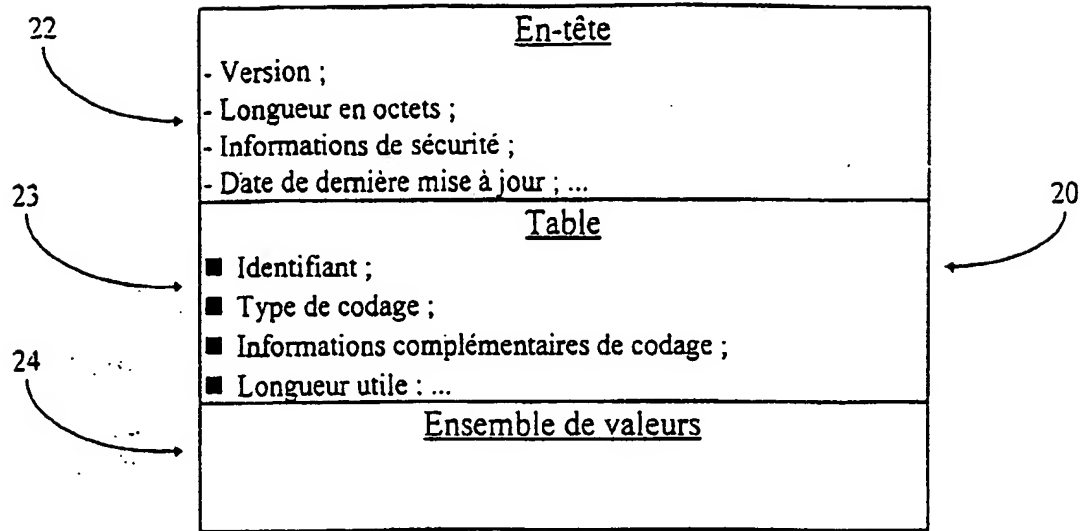


FIG.2

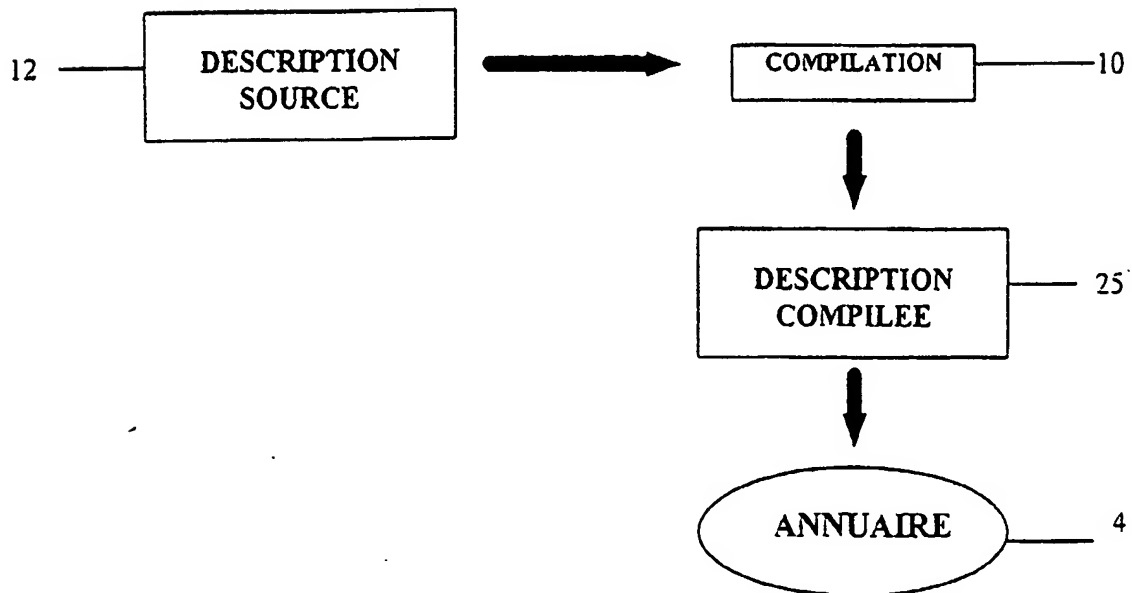


FIG.3



Office européen
des brevets

RAPPORT DE RECHERCHE EUROPEENNE

Numéro de la demande
EP 00 40 1375

DOCUMENTS CONSIDERES COMME PERTINENTS			
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes	Revendication concernée	CLASSEMENT DE LA DEMANDE (Int.Cl.7)
X	"Universal Data Option for Informix Dynamic Server" INFORMIX CORPORATION, 'en ligne! 1998, pages 1-8, XP002128177 Extrait de l'Internet: <URL:http://www.informix.com/informix/tech/briefs/udo/udo.pdf> 'extrait le 2000-01-14! * le document en entier *	1,2,11, 12,15	G06F17/30
A	REINWALD B ET AL: "STORING AND USING OBJECTS IN A RELATIONAL DATABASE" IBM SYSTEMS JOURNAL,US,IBM CORP. ARMONK, NEW YORK, vol. 35, no. 2, 1 mars 1996 (1996-03-01), pages 172-190, XP000594128 ISSN: 0018-8670 * page 175, colonne de gauche, ligne 29 - page 180, colonne de gauche, ligne 32 *	1-3,6-9, 11-15	
A	BLIUJUTE R ET AL: "Developing a DataBlade for a new index" PROCEEDINGS 15TH INTERNATIONAL CONFERENCE ON DATA ENGINEERING (CAT. NO.99CB36337), PROCEEDINGS OF IEEE COMPUTER SOCIETY 15TH INTERNATIONAL CONFERENCE ON DATA ENGINEERING, SYDNEY, NSW, AUSTRALIA, 23-26 MARCH 1999, pages 314-323, XP002131457 1999, Los Alamitos, CA, USA, IEEE Comput. Soc, USA ISBN: 0-7695-0071-4 * page 315, colonne de droite, alinéa 4 - page 320, colonne de gauche, alinéa 5.2 *	1,4-9, 11-15	DOMAINES TECHNIQUES RECHERCHES (Int.Cl.7) G06F
Le présent rapport a été établi pour toutes les revendications			
Lieu de la recherche LA HAYE		Date d'achèvement de la recherche 29 août 2000	Examineur Fournier, C
CATEGORIE DES DOCUMENTS CITES X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : arrière-plan technologique O : divulgation non-écrite P : document intercalaire		T : théorie ou principe à la base de l'invention E : document de brevet antérieur, mais publié à la date de dépôt ou après cette date O : cité dans la demande L : cité pour d'autres raisons S : membre de la même famille, document correspondant	